1st Install nginx

https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-20-04

Flow this command step by step:

```
1. sudo apt update
2. sudo apt install nginx
```

2nd Install MYsQL

https://www.vultr.com/pt/docs/how-to-install-mysql-5-7-on-ubuntu-20-04/

Flow this command step by step:

Add MySQL 5.7 APT Repository

1.wget https://dev.mysql.com/get/mysql-apt-config_0.8.12-1_all.deb

2.sudo dpkg -i mysql-apt-config_0.8.12-1_all.deb

3.
Then select **ubuntu bionic**
Then select **MySql Server & Cluster(8.0)**
Then select **mysql-5.7**
Last select **OK**

4.sudo apt update
5.sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 467B942D3A79BD29
5.sudo apt update
6.sudo apt-cache policy mysql-server

Install MySQL 5.7

```
1.sudo apt install -f mysql-client=5.7* mysql-community-server=5.7* mysql-server=5.7*

2.Then Choice password
```

Secure MySQL 5.7 Installation

```
1.sudo mysql_secure_installation
```

Check MySQL Version

```
1.mysql -u root -p
2.SELECT VERSION();
3.create database "name of database"
4.show databases;
5.exit;
```

3rd Install PHP

https://computingforgeeks.com/how-to-install-php-on-ubuntu-linux-system/

Flow this command step by step:

```
1.sudo apt update
2.sudo apt install lsb-release ca-certificates apt-transport-https software-properties-common -y
3.sudo add-apt-repository ppa:ondrej/php
4.sudo apt install php8.1
5.sudo apt install php8.1-fpm php8.1-common php8.1-mysql php8.1-xml php8.1-curl php8.1-gd php8.1-imagick php8.1-cli php8.1-imap php8.1-mbstring php8.1-opcache php8.1-zip
```

4th install Composer

```
1.curl -sS https://getcomposer.org/installer -o /tmp/composer-setup.php
2.HASH=`curl -sS https://composer.github.io/installer.sig`
3.php -r "if (hash_file('SHA384', '/tmp/composer-setup.php') === '$HASH')
{ echo 'Installer verified'; } else { echo 'Installer corrupt';
unlink('composer-setup.php'); } echo PHP_EOL;"
4.sudo php /tmp/composer-setup.php --install-dir=/usr/local/bin
--filename=composer
5.composer
```

5th Step

1.went to root directory cd ~
2. cd /var/www/html/
3.ll
4.Then git clone (project Clone with HTTPS)
5.Then (composer install)
6.cp .env.example .env
7.php artisan key:generate
8.nano .env
9. Then set database name and password and enter **(ctrl x)** and select **y** and **enter**

6th Setp

1.cd /etc/nginx/sites-available
2.comment out some line

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        # SSL configuration
        #
        # listen 443 ssl default_server;
        # listen [::]:443 ssl default_server;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
         # include snippets/snakeoil.conf;

        root /var/www/html/lemp_setup/public;

        # Add index.php to the list if you are using PHP
        index index.html index.php;

        server_name 188.166.220.195;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                        try_files $uri $uri/ /index.php?$query_string;
        }

        # pass PHP scripts to FastCGI server
        #
        location ~ \.php$ {
                include snippets/fastcgi-php.conf;

                 #With php-fpm (or other unix sockets):
                fastcgi_pass unix:/run/php/php8.1-fpm.sock;
                # With php-cgi (or other tcp sockets):
                #fastcgi_pass 127.0.0.1:9000;
        }

        # deny access to .htaccess files, if Apache's document root
        # concurs with nginx's one
        #
        location ~ /\.ht {
                deny all;
        }
}
```

```
3.sudo nginx -t
4.sudo systemctl reload nginx
```

Laravel Permissions

- Owner Permission: `sudo chown -R :www-data /var/www/html/project_folder`
- Storage Permission: `sudo chmod -R 775 /var/www/html/project_folder/storage`
- Cache File Permission: `sudo chmod -R 775 /var/www/html/project_folder/bootstrap/cache`